

PATENT
5625-00300

"EXPRESS MAIL" MAILING LABEL NO. EL726370081US
DATE OF DEPOSIT February 16, 2001

I HEREBY CERTIFY THAT THIS PAPER OR FEE IS BEING
DEPOSITED WITH THE UNITED STATES POSTAL SERVICE
"EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE
UNDER 37 C.F.R. §1.10 ON THE DATE INDICATED ABOVE
AND IS ADDRESSED TO THE ASSISTANT COMMISSIONER
OF PATENTS, WASHINGTON, D.C. 20231



Derrick Brown

**FRAME HANDLER FOR HIGH-SPEED
FIBER OPTIC COMMUNICATION INTERFACE**

By:

Hanan Potash

Atty. Dkt. No.: 5625-00300

Kevin L. Daffer/TAS
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, TX 78701-3102
Ph: (512) 703-1242

BACKGROUND OF THE INVENTION

Field of Invention

5 This invention relates to communications, and more particularly, to a high-speed fiber optic interface for communications.

Description of Related Art

10 In the near future, digital communications networks are expected to undergo tremendous growth and development. As their use becomes more widespread, there is an attendant need for higher bandwidth. To fill this need, present-day copper wire-based systems may gradually be replaced by fiber optic networks. Two key factors enabling this trend will be inexpensive fiber optic links and high-speed distributed protocol processors.
15 The latter are essential in preserving bandwidth while interfacing between dissimilar protocols, such as SONET, Ethernet, Fiber Channel and FireWire.

 The SONET (Synchronous Optical NETWORK) standard for digital optical transmission was introduced in the late 1980's to provide a means for digitally encoding
20 voice, video and other information for transmission over optical fiber. In a SONET communications network, digital information may be transmitted over an optical fiber by Pulse Code Modulation (PCM) of an optical signal. Analog signals (such as ordinary voice communications) must be digitized prior to being transmitted. For example, an analog telephone signal with a bandwidth of 3.1 KHz is sampled 8000 times per second,
25 quantized and encoded to byte samples and then transmitted at a bit rate of 64 Kbit/s. Because of the enormous bandwidth of the optical carrier, large amounts of information can be conveyed over a single fiber. To fully exploit this bandwidth, it is common to combine digital data from several sources by multiplexing them into a high-speed serial byte stream. The bit stream rate is high enough that the original data may be recovered at
30 the SONET receiver without distortion.

In SONET terminology, communication occurs between *nodes*. For example, a message may be transmitted from one node and received by other nodes. The basic unit of information exchanged between nodes is a discrete collection of bytes called a *frame*.

5 A frame always contains a prescribed number of bytes, and comprises *manifest* and *payload* portions. The payload represents the actual content, and the manifest (sometimes called the *transport overhead*, or *header*) contains frame synchronization bytes, pointers, addresses, and various other information required to recover the data within the payload. The payload portion of the frame is generally a composite of several different types of
10 signals. The constituent signals are merged via Time Division Multiplexing (TDM) into a single payload, and subsequently extracted from the payload at the destination node, using the information contained in the manifest. The so-called "primary rate" (T1 or DS1) used in the U.S., Canada and Japan uses frames consisting of 24 digitized analog voice channels, combined with the necessary signaling information.

15

SONET networks consist of various types of nodes linked by optical fiber. The simplest nodes are Repeaters, (also known as Regenerators). Repeaters buffer and retransmit incoming frames without altering their content, to overcome signal loss, timing skew, and jitter. Terminal Multiplexers are used to combine multiple input sources into
20 higher-level OC-n signals, or to perform the complementary decomposition of the incoming OC-n into constituent signals. For example, a Terminal Multiplexer may be used to combine three OC-1 inputs into a single OC-3, or to separate a single OC-1 into component DS-1 signals. A special type of multiplexer, known as an Add-Drop Multiplexer (ADM), inserts or removes constituent signals from a SONET frame without
25 affecting the rest of the payload. This, and the fact that the natural topology of fiber optic lines is a ring topology, makes it possible to configure multiple SONET nodes in a ring network.

A SONET network is described in terms of the Open System Integration (OSI) model of layers. The lowest layer is the *physical layer*, which represents the transmission medium; this is usually a set of fiber links. The next highest layer is the *section layer*, which is the path between repeaters. A portion of the manifest in each frame (the *section overhead*) is devoted to the signaling required for this layer. The third layer is the *line layer*, which refers to the path between multiplexers. The remainder of the manifest (the *line overhead*) comprises the signaling needed for the line layer. The highest layer is the *path layer*, which consists of the link of the SONET network from the point where the asynchronous digital signals enter to where these signals exit the SONET network.

10

The present SONET standard comprises various levels, based on the data rate. The following are the most commonly used:

STS-1/OC-1	51.84 Mbit/s
STS-3/OC-3	155.52 Mbit/s
STS-12/OC-12	622.08 Mbit/s
STS-48/OC-48	2488.32 Mbit/s
STS-192/OC-192	9953.28 Mbit/s

15 In this family of standards, the “STS-n” nomenclature refers to the “synchronous transport signal” and applies to the electrical signal, while the corresponding “OC-n” nomenclature refers to the associated optical signal. The basic format for all the information conveyed over a SONET network is the STS-1 frame, which consists of 810 bytes. The frames for all the higher levels are derived from combinations of STS-1 frames, either through multiplexing or simple concatenation. For example, an STS-3 frame, which is essentially composed of three STS-1 frames, consists of 2430 bytes. Note that, fundamental to the SONET definition, and enabled by the correspondingly higher bit rates, the frame rate for every STS-n/OC-n level is the same frame rate (preferably, 8000 frames per second.)

25

An increase in the number of users and a demand for greater bandwidth have resulted in the expansion of fiber optic-based communication networks and the move to higher data transmission rates. In the past, these data rates have been within the capabilities of traditional designs for multiplexers, routers, etc. However, with bit rates
5 as high as 10 Gbit/s, state of the art electronic circuitry is required within the network nodes. Thus, as the trend to higher data rates continues, the SONET electronics will become a limiting factor. Of course, ongoing improvements in semiconductor processing will yield faster transistors, which can extend the capabilities of traditional circuitry. But, as the amount and speed of communications traffic continues to far outpace circuit speed
10 improvements, architectural improvements in the electronic circuitry are fundamentally necessary.

As used herein, the term "frame handler" refers to a type of high-speed communications interface, operating between the fiberoptic backbone ring(s) and the
15 memory of the communications network interface card (NIC) or circuit switching ADM (Add Drop Multiplex) box. The frame handler moves payload traffic to and from the network, and must recognize and interpret any of various communications protocols present on the network. One limitation of common frame handlers is that they are "blocking" nodes. A blocking frame handler must capture an entire frame before it can
20 forward some or all of the payload. This becomes a disadvantage over long paths, where are a large number of blocking nodes are involved in relaying the communications traffic from the sender to the receiver. Assume, for instance, that the communication is a simple telephone conversation. Further assume that the signal must traverse 1000 blocking nodes along the path from the sender to the receiver. As stated above, the SONET frame
25 rate is 8000 frames per second, which implies that there is a 125 μ s delay associated with capturing the frame at each blocking node. Since each node must capture the entire frame before it is relayed to the next node, there is a cumulative delay of at least $1000 \times 125 \mu$ s, or 125 ms. A delay of this magnitude would be quite audible (and unpleasant) to a human listener. Furthermore, the high-speed memory required to implement a large frame buffer
30 is typically costly.

A further disadvantage of present SONET frame handlers is that they do not support broadcasting, but instead employ a point-to-point multicast method to distribute messages. Thus, in order for a sender connected in a ring to distribute a message to several other nodes within the ring, it is necessary to resend the message to each intended recipient. In view of these disadvantages, it would be desirable to have an efficient mechanism for broadcast communication, instead of only a multi-cast environment.

SUMMARY OF THE INVENTION

10

The problems outlined above are in large part solved by a system and method for a frame handler that interfaces a SONET communications network to a computer processor (i.e., optical to wired). According to this system and method, the frame handler operates in non-blocking mode -- i.e., it can perform add/drop modifications to an incoming STS-n frame and begin re-transmission of the frame before the last incoming byte of the frame is received. The latency of the frame handler is therefore much less than that of conventional designs, which buffer an entire frame before re-transmitting it. Cost is also reduced, by avoiding the requirement for large amounts of high-speed memory in which to store the frame.

20

In an embodiment of this system and method, STS- n frames are received and transmitted as a byte data stream. The incoming data stream is parsed "on the fly", and bytes are added to or removed from the STS- n frame as the data stream passes through the frame handler. This differs from conventionally designed frame handlers, in which the incoming serial data are collected until the entire STS- n frame has been buffered in memory, before parsing the frame.

The architecture of the frame handler is highly parallel and is configurable to handle various STS- n frame sizes (i.e., STS-1, STS-3, STS-12, STS-48, or STS-192), and communication protocols. The frame handler deals with higher-order STS- n frames

formed from interleaved lower-order frames (e.g., an STS-3 formed from the interleaved combination of three STS-1 frames). Counters, address pointers, etc. in the frame handler represent the relationship of each incoming byte to the constituent STS-1 frame with which it is associated. Thus, for example, the frame handler internally represents the fact that the 8th byte in an STS-3 frame originated as the 3rd byte in the second of three interleaved STS-1 frames.

The frame handler comprises symmetrical input and output sections, each consisting of a queue manager, pointer control logic, a frame index and an SPE index. The queue manager receives the incoming byte data stream, transfers data between the stream and the memory of the computer processor, and transmits the outgoing byte data stream. In an embodiment of the system and method disclosed herein, the queue manager comprises multiple data capture queues. Each of the data capture queues contains a scrambler/de-scrambler, operating in byte-parallel mode, and a capture table. The capture table contains flag bits, each of which denotes a desired byte within the synchronous payload envelope (SPE). In addition to the multiple data capture queues, the queue manager also contains an overhead capture queue. Analogously to the data capture queues, the overhead capture queue transfers data between the manifest portion of the STS- n frame and the memory of the computer processor. A "set word" instruction programs the queue manager to configure the data port queues according to parameters supplied in the instruction. This instruction includes a broadcast bit, which may be invoked for efficient multi-node distribution of a SONET frame. The broadcast bit induces the queue manager to copy, rather than remove, data from an incoming data stream to be transferred to the memory of the computer processor. When this is done, the frame may be forwarded directly to the next add/drop node in the network, without the sender having to retransmit the frame to each of the intended recipients.

The pointer control logic computes the offset to the start of the SPE within the STS-n frame. Included in the pointer control logic are a set of modulo counters, which compute the STS-1 frame number, and row/column position of each byte in the incoming

byte data stream. Also included in the pointer control logic is a frame bias table, which records these offsets. This table may be implemented using two-port random access memory (RAM), and shared between the input and output sections of the frame handler.

5 The frame index consists of logic to compute the location of the current incoming data byte within the STS- n frame. Similarly, the SPE index computes the logical address of the current incoming byte within the SPE. The system as disclosed herein is amenable to implementation as an integrated circuit.

10 A method for interfacing a SONET communications network to a computer processor or a circuit-switching box is also contemplated herein. The method includes receiving an STS- n as an incoming serial data byte stream, transferring data from the STS- n frame to the computer processor, modifying the STS- n payload by inserting data from the computer processor, and transmitting the modified payload in an outgoing STS-
15 n byte data stream. According to the method disclosed herein, the first byte of the outgoing payload data stream may be transmitted before the last byte of the incoming payload data stream is received. This amounts to non-blocking operation of the interface, since there is no requirement to buffer the entire received frame before it is re-transmitted.

20 Note, a new STS-n frame is constructed, by constructing a new manifest (header) and attaching to it the assigned payload at each transmission point. The frame is consumed (destroyed) at each reception point. The payload is passed on to the drop point or passed along to the transmit buffers.

25 The method further comprises queuing the incoming data stream in input capture queues and queuing the outgoing data stream in output capture queues; the plurality of capture queues accommodates multiple virtual tributaries within the STS-n frame. Overhead capture queues are also provided, for queuing incoming and outgoing data from
30 the manifest portion of the STS-n frame. Incoming data frames are de-scrambled for

reception and outgoing data frames are scrambled for transmission, using a scrambler/de-scrambler. The scrambler and de-scrambler operate in byte-parallel mode, since the serial bit rate of a 10 Gigabit transmission typically exceeds the speed of MOS logic gates. The method further comprises using a frame bias table to record the offset to the start of each
5 synchronous payload envelope (SPE) the payload proper within the STS-n frame, and a frame index to record the physical address of each byte in the incoming data stream with respect to the STS-n frame. The method also calls for an SPE index to record the address of each byte in the incoming data stream with respect to the SPE. Furthermore, a capture table is associated with each capture queue; the capture table contains flag bits, each of
10 which corresponds to a desired byte within the SPE.

In addition, the method disclosed herein provides for a *set port* instruction, which configures the capture queues according to parameters attached to the instruction. A broadcast bit in the set port instruction configures the frame handler for broadcasting. In
15 normal operation, data within the STS-n frame that are transferred to the computer processor are removed from the frame and replaced with a designated "Empty" byte before the frame is re-transmitted by the frame handler. On the other hand, in broadcast mode the data are copied, rather than removed. This is advantageous if there are several intended recipients for the data, since it allows the same frame to be forwarded to the next
20 add/drop node, and avoids the sender having to issue individual copies of the information to each intended recipient.

Note that it may be desirable to assign particular values to the "Empty" byte, depending on the situation. For example, clock regeneration circuitry within the frame
25 handler recovers the data clock rate from logic level transitions in the SONET bit stream. By assigning an alternating bit pattern such as 10101010 (xAA, in hexadecimal notation) to the Empty byte, clock recovery is made easier. On the other hand, the Empty byte may be assigned a pattern of all 0's or all 1's to deliberately stress the clock regeneration circuitry for diagnostic purposes.

30

For the reasons discussed above, the system and method described herein are believed to offer better performance than existing frame handlers, as well as being significantly less expensive.

5 While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and
10 alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in
5 which:

Fig. 1a shows the structure of a standard STS-1 frame;

Fig. 1b shows the transmit/receive byte order corresponding to the STS-1 frame
10 structure;

Fig. 2 illustrates how a single synchronous payload envelope (SPE) may span consecutive STS-1 frames;

Fig. 3 depicts byte-interleaving, by means of which, 3 STS-1 frames are combined
15 into a single STS-3 frame;

Fig. 4 contains a block diagram of the input section of an embodiment of the SONET frame handler disclosed herein;

Fig. 5 contains a block diagram detailing the internal structure of an embodiment of a capture queue and a parallel-mode $X^{43} + 1$ scrambler/de-scrambler; and
20

Fig. 6 contains a block diagram of the output section of an embodiment of the SONET frame handler disclosed herein.
25

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Data communications traffic is increasing at an explosive rate. This is due in part to growth in the number of transactions, as e-commerce and Internet use continue to rise.

5 A second factor has been an increase in the average size of the transactions, in terms of the sheer number of bytes transferred. This is evident as companies make greater use of teleconferencing and individuals routinely retrieve bulky multi-media files from websites. Optical data communications networks may potentially answer the demand for greater bandwidth. Current specifications for optical transport networks allow transmission rates
10 up to almost 1Gbps (OC-192) on a single optical wavelength. And future developments, such as dense wavelength division multiplexing (DWDM), which can presently transmit 128 wavelengths (colors) and holds forth the possibility of transmitting many more wavelengths (colors) over a single optical fiber, potentially extend this carrying capacity to over 800 Gbps.

15

To fully exploit this bandwidth, a high-speed interface is needed between the fiber optic channel conveying the data and the electronic data communications equipment for which the data is intended. Obviously, any such interface must comply with operative standards. The prevalent technology for telecommunications “backbones” (i.e., major
20 routes in a communications network) is based on the Synchronous Optical NETWORK (SONET) standard.

SONET data are transferred in “frames”. The structure of an STS-1 frame, the fundamental unit in SONET, is shown in Fig. 1a. Each frame consists of two parts, a
25 *manifest* 10 (“manifest - a list of passengers or cargo carried on a plane or a ship” also called the “header,” or “transport overhead”) and a *synchronous payload envelope* 12 (also known as the “SPE”, or simply, the “payload”). The STS-1 frame is comprised of 810 bytes, organized as 9 rows and 90 columns; the first 3 columns comprise the manifest and the remaining 87 columns comprise the SPE. Frames are transmitted or received as a
30 byte data stream 14. The bytes in the frame are sent row-by-row, from top to bottom, left

to right, as shown in Fig. 1b. As described above, each SONET interface (or “node”) contains a clock regeneration circuit, also known as a “phase lock loop”, necessary to properly reconstitute frames from the incoming serial bit stream. Also present is a highly accurate clock, synchronized with a diurnal standard clock used in the running of the chip’s processor and generating the outgoing frames. The internal clock is periodically adjusted, so that its average rate is the same as that of the diurnal clock. Obviously, no two SONET nodes can have identical clock frequencies. In fact, their clock frequency will vary slightly over a diurnal period. However, since both nodes are synchronized to the diurnal clock, the total number of pulses issued by both clocks within the diurnal period will be identical within a predetermined margin.

Thus, a SONET node is capable (through periodic checks with the diurnal clock) of adjusting its frequency as necessary to maintain synchronicity acceptable by the upstream node, as well as (through the clock regeneration circuit) accommodating an asynchronous incoming data stream. The nominal SONET frame transmission rate is 8000 frames per second. Therefore, the STS-1 data rate can be computed as:

$$\frac{810 \text{ bytes}}{\text{frame}} \times \frac{8 \text{ bits}}{\text{byte}} \times \frac{8000 \text{ frames}}{\text{second}} = \frac{51.84 \text{ Mbits}}{\text{second}}$$

The payload portion of the frame represents the actual content, and the manifest contains frame synchronization bytes, pointers, and various other information required to recover the data within the payload.

Although the SONET frames themselves are transmitted synchronously, they may be used to transport data that is received isochronously with the frame generation. The term isochronous refers to a process in which data must be delivered within certain time constraints. For example, multimedia streams require an isochronous transport mechanism to ensure that data is delivered as fast as it is displayed and to ensure that the audio is synchronized with the video. Isochronous may be contrasted with asynchronous, which applies to processes in which data streams may be broken by random intervals, and

synchronous processes, in which data streams can be delivered only at specific intervals. Thus, isochronous service is not as rigid as synchronous service, but not as lenient as asynchronous service. Isochronous data is often thought of as streaming data. Isochronous data is typically sent in real time without delays therein so that the data can also be played in real time. Isochronous data is therefore data that is sent in a regular, periodic and continuous fashion. Whereas, non-isochronous data is data that can be sent in bursts. It is therefore not important that non-isochronous data be sent in a continuous, periodic fashion.

The isochronous timing of SONET data may be understood through an analogy in which the SONET frame is represented by a bus that always departs on schedule, and the data by passengers who arrive at various times to board the bus. The bus leaves at its prescribed departure time, whether or not it is full, carrying however many passengers have boarded the bus in time. Passengers arriving too late to board the first bus must board the next one. In a similar fashion, isochronous data are incorporated into the synchronous SONET frame format. To accommodate the isochronous data, the payload is allowed to begin anywhere within the designated SPE portion of the frame, and will span to the next consecutive frame. This is illustrated in Fig. 2, where two consecutive frames **20** and **22** are used to transmit the first **26** and second **30** halves of a payload. The manifest **24** for frame L is comprised of bytes used for synchronization, status, parity, etc., for the SPE frame L whose first part is in STS frame L and second part in STS frame $L + 1$. The two halves of the payload and the associated manifest are contained within the heavy outline in Fig. 2.

Two bytes within the manifest, H_1 and H_2 **32**, constitute the payload pointer in frame L . Bytes H_1 and H_2 form a binary number from 0 to 782, indicating the starting position of the payload envelope (SPE) within the STS frame. The payload spans two consecutive frames, as shown in Fig. 2, and byte H_3 **34** becomes a "pointer action" byte, used to adjust pointer values in response to the effects of phase and frequency differences between the data and the frame. Because the incoming data are isosynchronous with

respect to the timing of the STS-1 frame, it may happen that the starting position of the payload must be adjusted. For example, if the data clock is advancing in phase (running slightly faster) with respect to the frame clock, eventually the data will arrive "too early". This is dealt with by stuffing an incoming data byte into the H_3 byte position 34 in manifest 28 of frame $L+1$, effectively advancing the frame timing. In a subsequent frame, the payload pointer 32 is adjusted (i.e., decremented) to "catch up" with the data. This mechanism allows SONET frame timing to dynamically adapt to phase and frequency changes in the data stream, and to the effects of variations in node-to-node timing.

10 Note that, a SONET node can begin to relay a payload after acquiring data from the first SPE, rather than having to wait until the entire SPE payload has been received. This greatly expedites the transfer of data from node to node in a SONET network.

15 The SONET standard allows for dynamic bandwidth allocation, to accommodate "bursty" communications traffic, with a very high peak-to-average bandwidth requirement. SONET specifies the following rate hierarchy:

STS-1/OC-1	51.84 Mbit/s
STS-3/OC-3	155.52 Mbit/s
STS-12/OC-12	622.08 Mbit/s
STS-48/OC-48	2488.32 Mbit/s
STS-192/OC-192	9953.28 Mbit/s

20 Conveniently, frames at lower transmission speeds can be multiplexed into higher rate frames, by a technique known as "byte-interleaved multiplexing". Thus, for example, three STS-1 frames can be combined into an STS-3 frame with no loss of data. This technique is illustrated in Fig. 3.

25 In Fig. 3, three STS-1 frames are merged, using byte-interleaved multiplexing, into a single STS-3 frame. As described above, an STS-1 frame is organized as 9 rows x 90 columns, with the first 3 columns of the frame containing the manifest, and

the other 87 columns the payload. An STS-3 frame has exactly three times the number of bytes as an STS-1 frame. Thus, the manifest of an STS-3 frame spans 9 byte columns, and its payload 261 columns. Byte-interleaved multiplexing populates the STS-3 frame by taking bytes from each of the three STS-1 frames in turn, preserving the order in which they occur in the STS-1 frame. In Fig. 3, the three constituent frames, **40**, **42** and **44**, are distinguished by a hatch pattern (i.e., no hatch, right-leaning hatch and left-leaning hatch). Also note that some of the bytes in all three STS-1 frames are numbered. The resultant STS-3 frame **46** retains these hatch patterns and byte numbering to indicate how the bytes from the three constituent frames are distributed. Referring to the upper left corner of STS-3 frame **46**, for example, it can be seen that the first byte in STS-3 frame **46** is taken from the first byte in STS-1 frame **40**, the second from the first byte in STS-1 frame **42**, and the third from STS-1 frame **44**. Similarly, the fourth byte in STS-3 frame **46** is taken from the second byte in STS-1 frame **40**, the fifth byte from the second byte in STS-1 frame **42**, etc. Note that, since the STS-3 frame rate is the same as that for the STS-1 frames (i.e., 8000 frames per second), the bit (and thus byte) rate is three times higher:

$$\frac{3 \times 810 \text{ bytes}}{\text{frame}} \times \frac{8 \text{ bits}}{\text{byte}} \times \frac{8000 \text{ frames}}{\text{second}} = \frac{155.52 \text{ Mbits}}{\text{second}}$$

“Bursty” communications traffic is characterized by a moderate average bit rate, with occasional high bit rates. It can be seen that byte-interleaved multiplexing can provide a means whereby a transitory demand for greater bandwidth can be satisfied, by formatting the incoming data into higher-order frames, thereby increasing the transmission rate.

SONET-compliant data transmission networks are often called upon to transmit asynchronous data formats, such as DS1 (with a capacity for 24 digitized voice-grade telephone signals) or ATM (“asynchronous transfer mode”, in which small fixed-size “cells” are used to transmit video, audio and computer data at up to 622 Mbps). SONET supports the embedding of data structures within the STS-X frame, through the use of “virtual tributaries”. There are four types of virtual tributaries specified for SONET

multiplexing: VT 1.5, VT 2, VT 3 and VT 6. Within the payload of the SONET frame, each of these is organized into a virtual tributary group (VT-G). The VT-G's are transmitted as 6.912 Mbps signals, seven of which can be multiplexed into an STS-1 frame. As an example, the VT 1.5 tributary carries 1.544 Mbps DS-1 signals, and four
5 VT 1.5's form a VT-G. Therefore, 28 VT 1.5 virtual tributaries can be encapsulated in a single STS-1 frame. Since each DS-1 represents 24 digitized voice-grade telephone channels, a total of 672 standard voice channels may be multiplexed into a single STS-1 frame.

10 A "frame handler" is a part of a SONET node situated between the fiber optic backbone and the communications network interface card (NIC) in a computer or a communication switch. The frame handler moves data to and from the SONET network, recognizes the beginning of an incoming frame, interprets the incoming manifest and constructs the outgoing frame and manifest according to the SONET communications
15 network protocols. At higher data communication rates the frame handler must process frames as expeditiously as possible, and with bit rates approaching 10 Gbps (for STS-192) the capabilities of conventional frame handlers may be strained. The options for increasing frame handler throughput are limited. One approach is simply to rely on advances in semiconductor technology to improve the performance of existing designs,
20 by increasing the operating speed of the individual components. However, progress in this area is difficult to predict, and it is uncertain whether faster devices will appear soon enough to be practical. A second approach is to design the frame handler so that frame-processing operations are shared by multiple conventional processors, working in parallel to enhance throughput. Here again, the processor architecture does not change – we
25 merely allocate more processors to the task. This strategy is made viable by the fact that the entire frame-processing task is readily subdivided into independent sub-tasks. Unfortunately, since this approach effectively multiplies the circuitry within the frame handler, it will add considerably to its cost.

The third approach to achieving higher bandwidth in the frame handler is to improve its basic design, making it inherently faster. This approach avoids dependence on yet to be developed semiconductor technology, and offers performance gains at a lower cost than highly parallel architectures. Among the advantages of the novel frame handler architecture disclosed herein are that it is "non-blocking".

Common frame handlers suffer from the disadvantage that they are "blocking" nodes -- i.e., they must capture an entire frame to internal memory before they can distribute the payload. This incurs a delay of $\frac{1}{8000} = 125 \mu\text{s}$ per node. Over a long path, where the communications traffic must propagate through thousands of blocking nodes, the cumulative delay may become intolerable. In addition, the frame buffers used in blocking frame handlers require large amounts of high-speed memory, which tends to make them expensive. In contrast, the frame handler architecture disclosed herein allows incoming data to be distributed before the entire frame is received (as described in more detail below). This substantially reduces the propagation delay through the frame handler, compared to a blocking node, and would be particularly advantageous for long, multi-node communication paths.

A further disadvantage of present SONET frame handlers is that they do not support "broadcasting," but instead employ a point-to-point multicast method to distribute messages. This is analogous to the way e-mail is distributed to a list of recipients. Instead of relaying the e-mail message from one recipient to the next, the mail server generally has to resend the mail to each recipient. A similar situation applies for a conventional frame handler in a SONET network configured as a ring. To distribute a message to several other nodes within the ring, it is necessary to resend the message to each intended recipient. In contrast, the frame handler architecture disclosed herein supports broadcasting. In this case, as the message is distributed to nodes along the ring, each node copies the message and passes it on. This avoids the need for the node acting as the message server to repeatedly resend the message.

30

As discussed earlier, SONET networks consist of various types of nodes linked by optical fiber. An Add-Drop Multiplexer (ADM) node inserts or removes constituent signals from a SONET frame without affecting the rest of the frame, thus permitting multiple ADM nodes to be configured in a ring network. The frame handler architecture disclosed herein represents a type of ADM node. In an embodiment described herein, the frame handler provides 16 input and 16 output data ports for conveying payload traffic to and from the fiber optic network. A 17th channel is also provided, for the transfer of manifest information.

In an embodiment disclosed herein, the frame handler architecture comprises four segments: the Frame Index, the Pointer Control, the SPE Index and the Queue Managers. Fig. 4 contains a block diagram of the input section of an embodiment of the frame handler. The following discussion refers to Fig. 4, and explains the operation of each of these segments.

The Frame Index segment of the frame handler comprises the components enclosed within the top-most rectangular region in Fig. 4. This circuitry generates the physical address of each incoming byte in the received STS-n SONET frame. (A similar circuit in the output section of the frame handler, discussed below, performs this function for the bytes in outgoing frames.) The Frame Index is programmed to operate in one of the standard OC modes (OC-1, OC-3, OC-12, OC-48, or OC-192), using the 8-bit count mask 50. This mask sets the modulus of modulo-1/3/12/48/192 counter 52 to 1, 3, 12, 48, or 192. The Frame Index is synchronized to the incoming data stream by reference to the binary pattern contained in the first two bytes (A₁ and A₂ in Fig. 2) in the manifest of the SONET frame. A byte stream containing the specific bit pattern A₁ = 11110110 and A₂ = 00101000 (or, A₁ = x-F6 and A₂ = x-28, in hexadecimal notation) is used to establish synchronization. This byte stream precedes the frame, and the Frame Index counts consecutive occurrences of this bit pattern in preparation to receive the incoming frame. For example, synchronization with an incoming OC-48 frame is achieved after the Frame Index recognizes a stream of 48 consecutive occurrences of xF6 followed by 48

consecutive occurrences of x28. Following the first two bytes of the manifest (A1 and A2) in the first frame received after the Frame Index has achieved synchronization, the modulo-1/3/12/48/192 counter **52** is set to 0, modulo-90 counter **54** is set to 2, and modulo-9 counter **56** is set to 0. From this point, these three counters keep track of the
5 location of the current byte within the STS-n frame. As incoming bytes from the SONET data stream are received, modulo-1/3/12/48/192 counter **52** indicates the STS-n frame number.

Recall from the earlier discussion of byte-interleaved multiplexing that an STS-n
10 frame comprises bytes from N STS-1 frames. Given the position of a byte within the STS-n frame, it is possible to determine which STS-1 frame the byte was taken from, as well as its position within the STS-1 frame.

$m(\text{mod } n) = \text{STS-1 frame from which byte } m \text{ in STS-}n \text{ frame was taken}$

15 $\text{int}\left(\frac{m}{n}\right) = \text{position in original STS-1 frame of byte } m \text{ in STS-}n \text{ frame}$

Assume for example, that we have an STS-48 frame, and we want to know which of the 48 constituent STS-1 frames byte 273 originally came from. To do this, we use the first expression above to calculate $273(\text{mod } 48) = 33$. To determine the location of byte 273
20 within STS-1 frame 33, we use the second expression above to calculate $\text{int}(273/48) = 5$. These arithmetic relationships allow simple logic circuits in the Frame Index to keep track of the original STS-1 frame, column and row for the current byte. In particular, the modulo-1/3/12/48/192 counter **52** effectively performs the above calculations on the bytes incoming STS-n frame. By definition, a modulo-N counter counts events from 0 to N-1,
25 and then “rolls over” – i.e., it resets to 0 and continues counting. At any time, the current value held in the counter may be read. Furthermore, the counter issues an output pulse when it rolls over. Assume counter **52** is configured as a modulo-48 counter, and that it counts incoming bytes from an STS-48 frame. At any point, the count held in counter **52** represents the STS-1 frame from which the current byte originated – this is the value
30 given by the first expression above. Every 48th byte causes the counter to roll over,

transmitting an overflow pulse to modulo-90 counter **54**. The number of overflow pulses issued by the modulo-48 counter **52** is equivalent to the second expression above. Referring to the previous example, by the time byte 273 was received, modulo-48 counter **52** would have rolled over (and clocked modulo-90 counter **54**) 5 times, and would hold a count of 33. This correctly indicates that byte 273 was originally byte 5 in STS-1 frame 33.

Thus, in composite frames, such as STS-3, STS-12, STS-48, and STS-192, modulo-1/3/12/48/192 counter **52** indicates which constituent STS-1 frame the current byte is associated with. Within a given STS-1 frame, modulo-90 counter **54** indicates the column and modulo-9 counter **56** indicates the row of the current byte. Modulo-1/3/12/48/192 counter **52** is incremented by byte clock **94** each time another byte is received by the frame handler. Modulo-90 counter **54** is incremented when the modulo-1/3/12/48/192, for example, when counter **52** reaches a count of 47, for an STS-48 frame. This is consistent with the standard sequence for byte interleaving (as described in connection with Fig. 3), in which successive bytes are taken from the same column in each of the constituent STS-1 frames, before starting the next column. Similarly, modulo-9 counter **56** is incremented each time modulo-90 counter **54** reaches 89.

The Pointer Control segment of the frame handler maintains the logical addresses of the N SPEs within the STS-n physical frame. It will be recalled that the SPE is the 783 byte “payload” portion of an 810 byte STS-1 frame, and that an SPE will typically span two consecutive STS-1 frames. The H_1H_2 pointer processor **58** is responsible for maintaining the offset to the SPE within the STS-n frame. These offsets are updated by the H_1H_2 pointer processor **58** and stored in SPE frame bias table and change history **60**. The frame bias table maintains up to 192 pointers (for an STS-192 frame). The frame bias change history is used when the SPE bias is changed. A new bias coming from the H_1H_2 pointer processor **58** must remain valid for three consecutive 125 μ s frame cycles before it is entered into the bias table **60**. Note therefore, that an OC-48 contains 48

independent individual SPEs, each with its own bias with respect to the main OC-48 frame. A frame bias table also exists in the output section of the frame handler (discussed below), to keep track of the SPE bias in outgoing STS-n frames. The two tables may be individually or jointly implemented using a 256 entry two-port RAM. In this case, the
5 read port of the two-port RAM is addressed by the modulo-1/3/12/48/192 counter **52**, and the address counter for the write port of the two-port RAM is used to update the H_1H_2 pointer processor in the output section.

The SPE Index portion of the frame handler generates the logical address of the
10 current byte within the SPE payload. This address is valid only when the current byte is actually a payload byte – otherwise, a “non SPE data byte inhibit” line is activated, to indicate that the current byte is part of the manifest. The modulo-783 counter **64** indicates the physical address of the current byte within the STS frame. This address could be derived from the values in the modulo-90 counter **54** and the modulo-9 counter
15 **56**, but it is simpler to compute it independently. The modulo-783 counter **64** is initialized to 0 when modulo-1/2/12/48/192 counter **52**, modulo-90 counter **54** and modulo-9 counter **56** are initialized to 0, 2, and 0, respectively. This corresponds to the address of byte H_3 in the first STS-n frame (see Fig. 2). Once it is initialized, modulo-783 counter **64** remains synchronized with modulo-90 counter **54** and modulo-9
20 counter **56**, and counts every incoming byte, unless inhibited by the “non SPE data byte inhibit” line. The logical address of a payload byte is derived from its physical address in the SPE by modulo-783 adder **66**, which adds the to the physical address the SPE bias from SPE frame bias table **60**.

25 The Queue Managers segment of the frame handler comprises the following functional portions, each of which is discussed below:

- (1) 16 payload port queues
- (2) a port queue for frame manifest information
- 30 (3) the SPE payload pass-through logic

Each of the 16 payload port queues comprises an 8-word (64-bit words) capture queue **74** and **86**, containing an $X^{43} + 1$ scrambler/de-scrambler. Scrambling transforms any bit sequence into one with enough logic level transitions to enable clock recovery.

5 Fig. 5 shows the internal structure of the capture queues, along with the $X^{43} + 1$ scrambler/de-scrambler. Incoming bytes are received by high-speed XOR circuit **100** over an 8-bit data bus. Each byte received is XORed with an 8-bit value derived from previous bytes by passing them through a series of 8-bit registers **102** and **104**. Note that, in contrast to conventional serial scrambler designs, this scrambler performs the
10 scrambling operations “byte-wise.” This greatly enhances the speed with which the scrambler can process incoming data.

Referring again to Fig. 4, in addition to the capture queue **74** and **76**, each payload port also contains a 16-bit mask and compare word **72** and **84**, and a 783-bit capture table
15 **70** and **82**. The upper 8 bits of the mask and compare word are a mask, using which, the address formed by the lower 8 bits is compared against the STS-n ID from modulo-1/3/12/48/192 counter **52**. The 783-bit capture table **70** and **82** contains a logic 1 for every byte in the SPE frame. A match on the STS frame and the bit in the SPE queue table will move the incoming byte to the input buffer. A similar match in the output
20 section of the frame handler moves a byte from the output buffer to the outgoing data stream. A match is detected using a frame-under-mask selection, given a desired virtual channel within the frame. The frame-under-mask compares the “Modulo 1/3/12/48/192” frame location against the desired virtual channel. For example, if one is receiving a full STS-1 (say, STS-1 number 17) out of an STS-192, the compare will be binary 00010001, the mask all 0’s, and the 783 bit table all 1’s. If, instead of the entire virtual channel, one
25 wanted just a single DS0 (i.e., one telephone line out of the STS-1), the table would contain only a single 1 among the 783 entries.

The payload port queue is set up by a *set port* instruction, containing the following parameters:

- 5 (a) A 16-bit word containing the STS ID and STS ID mask. If this mask is all 1's, the payload port queue logic selects a single STS out of an STS-n. For example, an individual STS-1 might be selected out of the 48 STS-1 frames interleaved into an STS-48.
- 10 (b) A table of 15 64-bit words, containing the 783-bit STE capture bit map. This bit map is used to select specific bytes within the STE frame corresponding to virtual tributaries, or in the case of ATM or IP traffic, to protocol-specific structure within the SPE.
- 15 (c) A 32-bit starting address of the data destination.
- (d) A 32-bit size of the data destination table.
- (e) A 32-bit word containing the number of bytes in the incoming record.
- 20 (f) A 32-bit multiplexing index for the destination table. The multiplexing index separates consecutively written (or read) words in memory, to facilitate their subsequent formatting into constituent STS-1 channels. For example, an STS-48 is comprised of 48 STS-1 channels. Placing the incoming data 48 words apart allows for later reformatting of the data into 48 tables, with each table containing the consecutive data from an STS-1 SPE frame.
- 25 (g) A format word, portions of which include 3-bits defining the number of bytes per word, a left/right justification flag for partial word fill, a 32/64-bit word flag, a broadcast bit, etc.
- 30

The port queue for frame manifest information **92** is similar to the data port queues, and is accompanied by a 16-bit mask and compare word **90** and a capture table **88**. There are two significant differences between these components and their counterparts in the data port queues:

- 40 (a) The table **88** contains 27 entries and is directed by the H₁H₂ pointer control section **58**, based on modulo-90 **54** and modulo-9 counters **56**.

- (b) The format word in the corresponding set port instruction contains an additional 2-bit manifest entry. The first bit indicates that the table is to be ignored during the first SONET frame. The second bit indicates that the M_1 byte, rather than the H_3 byte, should be captured on the last frame.

5

The SPE payload pass-through logic 78 comprises a 1 Kbyte or higher size buffer, which functions as an interface and rate matching circuit between the input and output sections of the frame handler. The buffer utilizes a two-port RAM, with separate input and output ports and independent pointers for incoming and outgoing SPE data. The pass-through logic also contains a mechanism 80 for inserting a "fill byte" entry, and a broadcast bit for each of the 16 data ports. In standard mode, the fill byte is placed into empty bytes in the SPE; in broadcast mode, the original byte is retransmitted.

15 The broadcast bit is used in connection with multi-drop, distributed communications. In a communications network having a ring topology, it is often necessary to distribute the same information to multiple recipients on the network. This is typically done in a point-to-point fashion, wherein each recipient (i.e., drop point) removes the desired information from a received frame before relaying it to the next drop point on the network. When the frame is returned to the original sender, the information is reinserted into the next frame to be transmitted, so that another intended recipient could access it. A more efficient method of distributing information to multiple recipients is broadcasting. A broadcast drop point copies, rather than removes, the information before re-transmitting the content of the SPE(s). The next intended recipient can then get the information without the original sender having to resend it. After the broadcast SPE has traversed the ring, it is returned to the original or rebroadcasting sender, who then removes the information.

25 A diagnostic mode bit is implemented in the pass-through hardware, which allows the frame handler to send an error notification to a processor if an attempt is made to write to a byte that doesn't contain the fill byte value. The pass-through logic also includes an $X^{43} + 1$ scrambler/descrambler, similar to those in the payload port queues.

All the information in the STS-n frame is scrambled for transmission and de-scrambled for reception, except, per SONET specifications, for the A₁ and A₂ bytes of the manifest. Encoder circuitry combines the clock and data into the bit stream sent to the laser transmitter or photodiode receiver. A variety of bit stream encoding formats can be used, including standard SONET NRZ encoding, 8b/10b encoding, NRZ with address marker, 64/66 encoding, and/or 8b/10b with address marker.

The output section of the frame handler, shown in Fig. 6, is similar to the input section. Modulo-count mask **120** programs the modulo-1/2/12/48/192 counter **122** for the current STS-n frame size. Modulo-90 **124** and modulo-9 **126** counters indicate the column and row of the current byte, as in the input section. Byte clock **128** drives the counters at a frequency derived from the incoming data stream. H₁H₂ pointer processor **130** maintains the offset to the SPE within the current STS-1 frame. The offset is stored in SPE frame bias table and change history **132**. This table keeps track of the SPE bias in outgoing STS-n frames, and is the counterpart to the SPE frame bias table **60** in the input section of the frame handler. The two tables may be individually or jointly implemented using a two-port RAM.

A modulo-783 counter **134** and modulo-783 subtractor **136** convert the byte location within the STS-n SPE to the equivalent STS-1 location. 783-bit output tables **140** and **146**, and 16-bit mask and compare words **142** and **148**, perform the same function as their counterparts (discussed above) in the input section of the frame handler. Packet align **144** and **150** are used by the controlling program to delete fill bytes that were originally inserted in the incoming data files, typically in byte stream to word blocks conversion, or in other operations done by protocols such as ATM or HDLC that deposited the ADD data in main memory. For example, specific ATM 53 byte packets are stored in 7 64-bit words (56 bytes per packet); this section will remove the last three fill bytes before transmission. 8-word output capture queues **152** and **154** queue data from the 64-bit memory bus before transmission. Similarly, 27-bit table **164**, 16-bit mask and compare word **166**, and 8-word overhead output queue **168** format the manifest

before transmitting the STS-n frame. SPE payload pass-through logic **160** performs rate matching and buffering, and input clock to system clock byte hand over from the frame handler input section. The pass-through logic **160** and "fill byte" entry insertion mechanism **162** for the output section function similarly to their counterparts in the frame handler input section, discussed above.

The present system and method exploit novel architectural features to achieve performance and cost advantages over conventional frame handlers. Among these, is the ability to operate in non-blocking mode. As explained above, a traditional frame handler must capture the entire incoming STS-n frame, before parsing and re-transmitting it. Since the STS-n frame rate is 8 KHz, a delay of 125 μ s occurs at each blocking node through which the frame passes. In a large network, with many nodes, the cumulative delay may be intolerable. Furthermore, the need to buffer an entire STS-n frame in memory (155,520 bytes, for STS-192) means that traditional frame handlers require large amounts of high-speed RAM, which tends to make them expensive. In contrast, a frame handler embodying the system and methods disclosed herein parses the STS-n frame "on the fly" -- i.e., before the entire frame has been received by the frame handler. Consequently, the latency is much lower than for a conventional frame handler, reducing cumulative delay effects. A further benefit of the present system and method is the avoidance of extensive high-speed memory. This lowers the cost of the frame handler, compared to a conventional approach.

A preferred embodiment of the system and method disclosed herein is a low-cost high-performance frame handler, suitable for PC-based network applications. In such an embodiment, the frame handler would be integrated into the Network Processing Unit (NPU) a communication protocol processing chip and/or into a network interface card (NIC) installed within a network server, a workstation, or desktop computer. The ability of the frame handler to deal with various communications protocols, including embedded asynchronous protocols (such as ATM), recommends it for use in inexpensive network

interface hardware, allowing individual PC users to access high-bandwidth voice, video, and data communications.

It will be appreciated by those skilled in the art having the benefit of this disclosure that this invention is believed to present a system and method for interfacing a SONET communications network to a computer processor and/or to a communication switch node. Further modifications and alternative embodiments of various aspects of the invention will be apparent to those skilled in the art in view of this description. Such details as the number and depth of the data queues as described herein are exemplary of a particular embodiment, and may be altered in other embodiments. It is intended that the following claims be interpreted to embrace all such modifications and changes and, accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.